# Torndb

***Release 0.3***

**Aug 30, 2017**

# Contents


**1 Release history** **3**
- 1.1 Version 0.3, Jul 25 2014 . . . 3
- 1.2 Version 0.2, Dec 22 2013 . . . 3
- 1.3 Version 0.1, Sep 16 2012 . . . 3

**2 Indices and tables** **5**

**Python Module Index** **7**

A lightweight wrapper around MySQLdb.

Originally part of the Tornado framework. The tornado.database module is slated for removal in Tornado 3.0, and it is now available separately as torndb.

**class** torndb.**Connection**(*host*, *database*, *user=None*, *password=None*, *max_idle_time=25200*, *connect_timeout=0*, *time_zone='+0:00'*, *charset='utf8'*, *sql_mode='TRADITIONAL'*, ***kwargs*)

A lightweight wrapper around MySQLdb DB-API connections.

The main value we provide is wrapping rows in a dict/object so that columns can be accessed by name. Typical usage:

```
db = torndb.Connection("localhost", "mydatabase")
for article in db.query("SELECT * FROM articles"):
    print article.title
```

Cursors are hidden by the implementation, but other than that, the methods are very similar to the DB-API.

We explicitly set the timezone to UTC and assume the character encoding to UTF-8 (can be changed) on all connections to avoid time zone and encoding errors.

The sql_mode parameter is set by default to "traditional", which "gives an error instead of a warning" (http://dev.mysql.com/doc/refman/5.0/en/server-sql-mode.html). However, it can be set to any other mode including blank (None) thereby explicitly clearing the SQL mode.

Arguments read_timeout and write_timeout can be passed using kwargs, if MySQLdb version >= 1.2.5 and MySQL version > 5.1.12.

**close**()
Closes this database connection.

**reconnect**()
Closes the existing database connection and re-opens it.

**iter**(*query*, **parameters*, ***kwparameters*)
Returns an iterator for the given query and parameters.

**query**(*query*, **parameters*, ***kwparameters*)
Returns a row list for the given query and parameters.

**get**(*query*, **parameters*, ***kwparameters*)
Returns the (singular) row returned by the given query.

If the query has no results, returns None. If it has more than one result, raises an exception.

**execute**(*query*, **parameters*, ***kwparameters*)
Executes the given query, returning the lastrowid from the query.

**execute_lastrowid**(*query*, **parameters*, ***kwparameters*)
Executes the given query, returning the lastrowid from the query.

**execute_rowcount**(*query*, **parameters*, ***kwparameters*)
Executes the given query, returning the rowcount from the query.

**executemany**(*query*, *parameters*)
Executes the given query against all the given param sequences.

We return the lastrowid from the query.

**executemany_lastrowid**(*query*, *parameters*)
Executes the given query against all the given param sequences.

We return the lastrowid from the query.

**executemany_rowcount**(*query*, *parameters*)
Executes the given query against all the given param sequences.

We return the rowcount from the query.

**update**(*query*, **parameters*, ***kwparameters*)
Executes the given query, returning the rowcount from the query.

**delete**(*query*, **parameters*, ***kwparameters*)
Executes the given query, returning the rowcount from the query.

**updatemany**(*query*, *parameters*)
Executes the given query against all the given param sequences.

We return the rowcount from the query.

**insert**(*query*, **parameters*, ***kwparameters*)
Executes the given query, returning the lastrowid from the query.

**insertmany**(*query*, *parameters*)
Executes the given query against all the given param sequences.

We return the lastrowid from the query.

**class** torndb.**Row**
A dict that allows for object-like property access syntax.

CHAPTER 1

# Release history

## Version 0.3, Jul 25 2014

- Added `charset` and `sql_mode` arguments to *`Connection`* constructor.

## Version 0.2, Dec 22 2013

- Query methods now accept keyword arguments and `%(name)s`-style formatting.
- New aliases (`insert`, `insertmany`, `update`, and `updatemany`) are available for the `execute` family of methods.
- The *`Connection`* constructor now takes `connect_timeout` and `time_zone` arguments.

## Version 0.1, Sep 16 2012

- Initial separate release, copied from Tornado 2.4.

CHAPTER 2

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## t

torndb, 3

# Index


## C

close() (torndb.Connection method), 1
Connection (class in torndb), 1

## D

delete() (torndb.Connection method), 2

## E

execute() (torndb.Connection method), 1
execute_lastrowid() (torndb.Connection method), 1
execute_rowcount() (torndb.Connection method), 1
executemany() (torndb.Connection method), 1
executemany_lastrowid() (torndb.Connection method), 1
executemany_rowcount() (torndb.Connection method), 1

## G

get() (torndb.Connection method), 1

## I

insert() (torndb.Connection method), 2
insertmany() (torndb.Connection method), 2
iter() (torndb.Connection method), 1

## Q

query() (torndb.Connection method), 1

## R

reconnect() (torndb.Connection method), 1
Row (class in torndb), 2

## T

torndb (module), 1

## U

update() (torndb.Connection method), 2
updatemany() (torndb.Connection method), 2